Detection of Selfish node in Replica allocation for improving Data accessibility in MANET

S. Geethanjali[1] K. Nagendran[2]
[1]Final Year M.E(CSE) [2]Assistant Professor
[1,2] Selvam College of Technology, Namakkal, India

Abstract— In a mobile ad hoc network, the mobility and resource constraints of mobile nodes may lead to network partitioning or performance degradation. Several data replication techniques have been proposed to minimize performance degradation. Most of them assume that all mobile nodes collaborate fully in terms of sharing their memory space. In reality, however, some nodes may selfishly decide only to cooperate partially, or not at all, with other nodes. These selfish nodes could then reduce the overall data accessibility in the network. In this paper, we examine the impact of selfish nodes in a mobile ad hoc network from the perspective of replica allocation. In particular, we develop a selfish node detection algorithm that considers partial selfishness and novel replica allocation techniques to properly cope with selfish replica allocation. Network partitions can occur frequently, since nodes move freely in a MANET, causing some data to be often inaccessible to some of the nodes. Hence, data accessibility is often an important performance metric in a MANET. Data are usually replicated at nodes, other than the original owners, to increase data accessibility to cope with frequent network partitions. A considerable amount of research has recently been proposed for replica allocation in a MANET. In general, replication can simultaneously improve data accessibility and reduce query delay, i.e., query response time, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the original data.

Key Terms: Mobile Ad hoc networks, Selfish replica allocation.

I. INTRODUCTION

In mobile computing environments, by utilizing wireless networks, users equipped with portable computers, called mobile hosts, can change their locations while retaining network connections. As one of the research fields in mobile computing environments, there has been an increasing interest in ad hoc networks which are constructed by only mobile hosts[3], [6]. In ad hoc networks, every mobile host plays the role of a router, and communicates with each other. Even if the source and the destination mobile hosts are not in the communication range of the two mobile hosts, data packets are forwarded to the destination mobile host by relaying transmission through other mobile hosts which exist between the two mobile hosts. Since no special infrastructure is required, in various fields such as military affair and commerce, many applications are expected to be developed in ad hoc networks. In ad hoc networks, since mobile hosts move freely, disconnections occur frequently, and this causes frequent network division. Consequently, different fundamental technologies from the conventional fixed networks are needed. For example, if a network is divided into two networks due to the migrations of mobile hosts, mobile hosts in one of the divided two networks cannot access data items held by mobile hosts in the other network. Thus, data accessibility in ad hoc networks is lower than that in the conventional fixed networks. Replication is very effective for improving the data accessibility and reduces query delay, i.e., query response time, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the original data. Each node in a MANET has resource constraints, such as battery and storage limitations. The selfish nodes may not transmit data to others to conserve their own batteries. Although network issues are important in a MANET, replica allocation is also crucial, since the ultimate goal of using a MANET is to provide data services to users. A selfish node may not share its own memory space to store replica for the benefit of other nodes. It is easily find such cases in a typical peer-to-peer application.

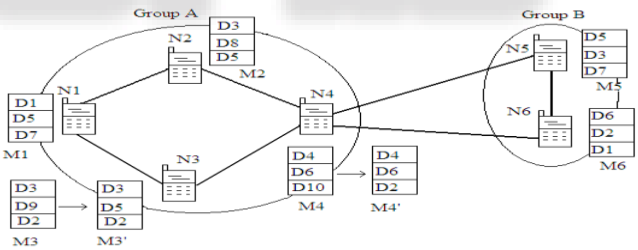

Fig .1 Example of selfish replica allocation

The Fig. 1 illustrates a replica allocation scheme, DCG [8], where nodes N1, N2, N6 maintain their memory space M1, M2, M6, respectively, with the access frequency information. In this, a straight line denotes a wireless link, a gray rectangle denotes an original data item, and a white rectangle denotes a replica allocated. As shown in Fig. 1, DCG seeks to minimize the duplication of data items in a group to achieve high data accessibility. Let us consider the case where N3 behaves “selfishly” by maintaining M’3, instead of M3, to prefer the locally frequently accessed data for low query delay. In the original case, D3, D9, and D2 were allocated to N3. However, due to the selfish behavior, D3, D5, and D2, the top three most locally frequently accessed items, are instead maintained in local storage. Thus, other nodes in the same group, i.e., N1, N2, and N4, are no longer able to access D9.This showcases degraded data accessibility, since N1, N2, and N4 cannot fully leverage N3's memory space as intended in cooperative replica sharing. As another example, a node may be only "partially selfish" in a MANET. For instance, node N4 may want to locally hold D2, one of the locally frequently accessed data items. In this case, N4 uses only a part of its storage for its own frequently accessed data items. In this case, N4 uses only a part of its storage for its own frequently accessed data, while the remaining part is for the benefit of overall data accessibility. Thus, N4 may decide to maintain M'4, instead of M4. Even with only partial selfishness, data accessibility is still degraded, since the other nodes in the same group, i.e., N1, N2, and N3, cannot access D10.

The partially selfish nodes (e.g., N4 in Fig. 1) should also be taken into account, in addition to the fully selfish nodes (e.g., N3 in Fig. 1), to properly handle the selfish replica allocation problem. We therefore need to measure the "degree of selfishness" to appropriately handle the partially selfish nodes. Motivated by this concept of "partial selfishness," we borrow the notion of credit risk (CR) [11] from economics to detect selfish nodes. Since the credit risk is calculated from several selfishness features in this paper, it can measure the degree of selfishness elaborately. In our scheme, a node can measure the degree of selfishness of another node, to which it is connected by one or multiple hops in a MANET. The novel replica allocation techniques with the developed selfish node detection method. They are based on the concept of a self-centered friendship tree (SCF-tree) and its variation to achieve high data accessibility with low communication cost in the presence of selfish nodes. The SCF-tree is inspired by our human friendship management in the real world. In the real world, a friendship, which is a form of social bond, is made individually [5]. For example, although A and B are friends, the friends of A are not always the same as the friends of B. With the help of SCFtree, we aim to reduce the communication cost, while still achieving good data accessibility.

II. NODE BEHAVIOR MODEL

The work considers only binary behavioral states for selfish nodes from the network routing perspective: selfish or not (i.e., forwarding data or not). It is necessary to further consider the partial selfish behavior to handle the selfish replica allocation. Therefore, we define three types of behavioral states for nodes from the viewpoint of selfish replica allocation2:

– *Type-1 node:* The nodes are nonselfish nodes. The nodes hold replicas allocated by other nodes within the limits of their memory space.

– *Type-2 node:* The nodes are fully selfish nodes. The nodes do not hold replicas allocated by other nodes, but allocate replicas to other nodes for their accessibility.

– *Type-3 node:* The nodes are partially selfish nodes. The nodes use their memory space partially for allocated replicas by other nodes. Their memory space may be divided logically into two parts: selfish and public area. These nodes allocate replicas to other nodes for their accessibility.

The detection of the type-3 nodes is complex, because they are not always selfish. In some sense, a type-3 node might be considered as nonselfish, since the node shares part of its memory space. In this paper, however, we have considered it as (partial) selfish, because the node also leads to the selfish replica allocation problem, as described in Section 1. Note that selfish and nonselfish nodes perform the same procedure when they receive a data access request, although they behave differently in using their memory space.

III. PROPOSED STRATEGY

Our strategy consists of three parts: 1) detecting selfish nodes, 2) building the SCF-tree, and 3) allocating replica. At a specific period, or relocation period [8], each node executes the following procedures:

– Each node detects the selfish nodes based on credit risk scores.

– Each node makes its own (partial) topology graph and builds its own SCF-tree by excluding selfish nodes.

– Based on SCF-tree, each node allocates replica in a fully distributed manner.

The CR score is updated accordingly during the query processing phase. We borrow the notion of credit risk from economics to effectively measure the "degree of selfishness." In economics, credit risk is the measured risk of loss due to a debtor's nonpayment of a loan. A bank examines the credit risk of an applicant prior to approving the loan. The measured credit risk of the applicant indicates if he/she is creditworthy. We take a similar approach. A node wants to know if another node is believable, in the sense that a replica can be paid back, or served upon request to share a memory space in a MANET. With the measured degree of selfishness, we propose a novel tree that represents relationships among nodes in a MANET, for replica allocation, termed the SCF-tree. The SCF-tree models human friendship management in the real world. The key strength of the SCF-tree-based replica allocation techniques is that it can minimize the communication cost, while achieving high data accessibility. This is because each node detects selfishness and makes replica allocation at its own discretion, without forming any group or engaging in lengthy negotiations.

1) *Detecting Selfish Node*

The notion of credit risk can be described by the following equation:

Credit Risk = expected risk/expected value

In our strategy, each node calculates a CR score for each of the nodes to which it is connected. Each node shall estimate the "degree of selfishness" for all of its connected nodes based on the score. We first describe selfish features that may lead to the selfish replica allocation problem to determine both expected value and expected risk.

2) *Building SCF-Tree*

The SCF-tree based replica allocation techniques are inspired by human friendship management in the real world, where each person makes his/her own friends forming a web and manages friendship by himself/herself. He/she does not have to discuss these with others to maintain the friendship. The decision is solely at his/her discretion. The main objective of our novel replica allocation techniques is to reduce traffic overhead, while achieving high data accessibility. If the novel replica allocation techniques can allocate replica without discussion with other nodes, as in a human friendship management, traffic overhead will decrease.Since the SCF-tree consists of only nonselfish nodes, we need to measure the degree of selfishness to apply real-world friendship management to replica allocation in a MANET.

3) Allocating Replica

After building the SCF-tree, a node allocates replica at every relocation period. Each node asks nonselfish nodes within its SCF-tree to hold replica when it cannot hold replica in its local memory space. Since the SCF-tree based replica allocation is performed in a fully distributed manner, each node determines replica allocation individually without any communication with other nodes. Since every node has its own SCF-tree, it can perform replica allocation at its discretion. Replicas are allocated based on the access frequency of group members. Each node executes this process at every relocation period after building its own SCF-tree. At first, a node determines the priority for allocating replicas. The priority is based on Breadth First Search.

A set of replica allocation techniques, as follows:

1. SCF-tree-based replica allocation (SCF): This technique is described in Algorithm 4 and serves as a basic SCF-tree based technique.
2. SCF-tree based replica allocation with degree of selfishness (SCF-DS): This technique takes into account the degree of selfishness in allocating replicas. That is, less selfish nodes should be visited first at the same SCF-tree level. This policy makes more frequently accessed data items reside on less selfish nodes.
3. SCF-tree based replica allocation with closer node (SCF-CN): This technique allocates more replicas to the closer nodes in the SCF-tree. That is, more replicas are allocated to the node with lower depth within the SCF-tree.
4. Extended SCF-tree based replica allocation (eSCF): This technique is based on an extended SCF-tree (eSCF-tree). In this technique, Ni builds its eSCFtree based on Gi, not Gns i . Consequently, eSCF-tree includes selfish nodes, as well as nonselfish nodes. Ni marks the detected selfish nodes within its eSCFtree and allocates replicas to the nonselfish nodes in its eSCF-tree.

IV. PERFORMANCE EVALUATION

A. Simulation Environment

In the simulation, the number of mobile nodes is set to 40.Each node has its local memory space and moves with a velocity from 0 ~1 (m/s) over 50 (m) _ 50 (m) flatland. The movement pattern of nodes follows the random waypoint model [5], where each node remains stationary for a pause time and then it selects a random destination and moves to the destination. After reaching the destination, it again stops for a pause time and repeats this behavior. The radio communication range of each node is a circle with a radius of 1~19 (m).Suppose that there are 40 individual pieces of data, each of the same size. In the network, node N_i ($1 \le i \le 40$) holds data D_i as the original. The data access frequency is assumed to follow *Zipf* distribution. The default relocation period is set to 256 units of simulation time which we vary from 64 to 8,192 units of simulation time. Table 1 describes the simulation parameters. The default number of selfish nodes is set to be 70 percent of the entire nodes in our simulation, based on the observation of a real application [1]. Set 75 percent of selfish nodes to be type-3 (i.e., partially selfish) and the remaining to be type-2 (i.e., fully selfish). Type-3 nodes consist of three groups of equal size. Each group uses 25, 50, and 75 percent of its memory space for the selfish area.

Parameter (unit)	Value (default)
Number of nodes	40
Number of data items	40
Radius of communication range(m)	1~19(7)
Size of the network(m)	50×50
Size of memory space(data items)	2~40(10)
Percentage of selfish nodes (%)	0~100(70)
Maximum velocity of a node(m/s)	1
Relocation period	64 ~ 8,192(256)
Zipf parameter	0.8

Table. 1:simulation parameters

Type-2 nodes will not accept replica allocation requests from other nodes in the replica allocation phase, thus being expected to create significant selfishness alarm in query processing. Type-3 nodes will accept or reject replica allocation requests according to their local status, thereby causing some selfishness alarms in subsequent query processing.

Evaluate our strategy using the following four performance metrics:

1. Overall selfishness alarm: This is the ratio of the overall selfishness alarm of all nodes to all queries that should be served by the expected node in the entire system.

2. Communication cost: This is the total hop count of data transmission for selfish node detection and replica allocation/relocation, and their involved information sharing.

3. Average query delay: This is the number of hops from a requester node to the nearest node with the requested data item. If the requested data item is in the local memory of a requester, the query delay is 0.
We only consider successful queries, i.e., it is the total delay of successful requests divided by the total number of successful requests.

4. Data accessibility: This is the ratio of the number of successful data requests to the total number of data requests. During 50,000 units of simulation time, we simulate and compare the proposed replica allocation strategies (i.e., SCF,SCF-DS, SCF-CN, and eSCF) with the following techniques:

– Static Access Frequency (SAF) [8]: Each node allocates replica based only on its own access frequency, without considering or detecting selfish nodes. This allocation technique is expected to show the optimal performance in terms of communication cost, because the technique does not communicate with others to allocate replica.

– Dynamic Connectivity-based Grouping (DCG)[8]: DCG creates groups of nodes that are biconnected components in a network, without considering or detecting selfish nodes. In each group, the node, called coordinator, allocates replicas based on the access frequency of the group. This technique is known to have high data accessibility.

– Dynamic Connectivity-based Grouping with detection (DCG$^+$): The technique combines DCG with our detection method. Initially, groups of nodes are created according to the DCG methodology. Subsequently, in each group, selfish nodes are detected based on our detection method. For the detection,
each node in a group sends its nCR scores to the coordinator with the lowest suffix of node identifer in the group. The coordinator excludes selfish node(s) from the group for replica allocation. As a result, only nonselfish nodes form a group again. The replica allocation is only performed within the final group without any selfish nodes. After replica allocation, the coordinator shares the information of replica allocation with group members for the subsequent selfishness detection. In particular, selfish nodes are determined to be selfish only when all other nodes in the group agree with the node's selfishness. We experimented with other approaches to determine selfishness, including the agreement of 1) at least one and 2) the majority of nodes. We chose to use the agreement of all other nodes experimentally, since this shows the best data accessibility performance in our experiments: our analysis reveals that at least one node approach shows

1) Parameter Setting in Our Strategy

Several parameters are used in this strategy. SCF-tree experimentally after inspecting our simulation results. We observe that average query delay, data accessibility, and communication cost are insensitive to the depth of the SCF-tree. More specifically, both average query delay and data accessibility are almost the same with varying depths of SCF-tree, while communication cost increases marginally as the depth increases.

1. Effectiveness of Detection Method

The overall selfishness alarm of DCG with that of DCG$^+$ to demonstrate the effectiveness of our detection method. Tthe overall selfishness alarm will be reduced in query processing by detecting selfish nodes effectively with DCG$^+$, since many selfish nodes will be removed from the replica allocation phase and many reliable nodes will serve data requests from nodes. However, recall that the selfishness alarm may also occur due to network disconnections, i.e., false alarm. Actually, it is desirable to observe truly selfish nodes to evaluate the effectiveness of the detection method. the worst data accessibility, whereas its communication cost and average query delay are marginally better than that of the others.

2. Communication Cost

The replica allocation techniques used in communication cost. Communication cost of DCG, DCG$^+$ and our techniques decreases with more selfish nodes, since the cost in fetching replicas and/or in group communication will be reduced. In the DCG technique, the effective memory space in the entire system gets reduced due to many selfish nodes, resulting in reduced cost in fetching replicas. In the DCG$^+$ technique, cost reduction is faster than in DCG, since fewer nodes participate in replica allocation. Note that the communication cost of our techniques is relatively stable. Communication reduction factor is much less than in DCG and DCG$^+$ and the distance between nonselfish nodes increases simultaneously.

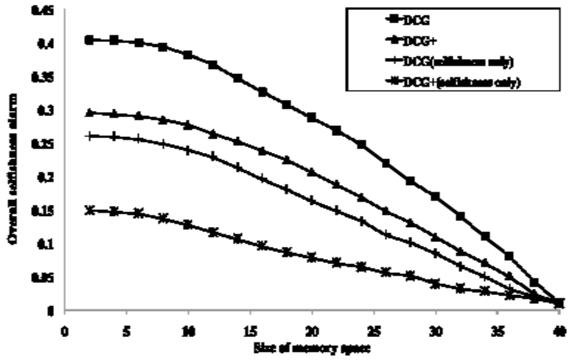

Fig. 2 Varying size of memory space

3. Average Query Delay

The eSCF technique shows the best average query delay. In the eSCF technique, nearby selfish nodes can be added to the eSCF-tree. Consequently, some queries are possibly served by the nearby (partially) selfish nodes, whereas only nonselfish nodes, which maybe far away, serve queries in other techniques. Our intuition was that query delay decreases as the size of memory space increases. As the size of memory space increases, many nodes will accept replica allocation/relocation requests, since the size of public memory space increases as well. As a result, more queries are served by nearby nodes or locally.

4. Data Accessibility

The techniques perform significantly better than other techniques in the presence of selfish nodes. In all cases, our techniques outperform SAF, DCG, and DCGþ considerably, since our techniques can detect and handle selfish nodes in replica allocation effectively and efficiently. Among our techniques, the eSCF technique shows a slightly poorer performance. Our initial intuition was that, data accessibility is stable with relocation periods.The performance of our techniques improves faster than do others, since our techniques fully utilize the memory space of nodes. The robustness of our techniques with respect to varying percentage of selfish nodes. The profit of DCG technique is considerably hampered by selfish nodes, whereas the SAF technique is insensitive at all.

5. Effect of Communication Range

The average query delay of all techniques degrades as the communication range increases, but it improves from a certain point, since when the communication range is larger than, the number of hops among connected nodes decreases. Data accessibility improves with the wide range of communication, since more nodes become connected. Clearly, our techniques work best.

V. RELATED WORK

A. Selfish Nodes from a Network Perspective

MANETs are divided into two categories: closed and open in the work [4] , [16]. In a closed MANET, all nodes voluntarily participate in and organize the network.However, in an open MANET, which we consider in this paper, however, individual nodes may have different objectives. In this case, some nodes can be selfish topreserve their own resources. Various techniques have been proposed to handle the problem of selfish behavior from the network perspective. As described in [16], techniques handling selfish nodes can be classified into three categories: reputation-based, credit-payment, and game theory-based techniques. In reputation-based techniques, each node observes the behaviors of others and uses the acquired information for routing [9], [10]. In credit-payment techniques, each node gives a credit to others, as a reward for data forwarding [2], [30]. The acquired credit is then used to send data to others. The game theory-based techniques assume that all rational nodes can determine their own optimal strategies to maximize their profit [7], [14]. The game theory-based techniques want to find the Nash Equilibrium point [13] to maximize system performance. All these techniques focused on packet forwarding. In contrast, this paper focuses on the problem of selfish replica allocation.

VI. CONCLUSION

The selfish nodes from the replica allocation perspective. We term this problem selfish replica allocation. The work was motivated by the fact that a selfish replica allocation could lead to overall poor data accessibility in a MANET. A selfish node detection method and novel replica allocation techniques to handle the selfish replica allocation appropriately. The proposed strategies are inspired by the real-world observations in economics in terms of credit risk and in human friendship management in terms of choosing one's friends completely at one's own discretion. The notion of credit risk from economics to detect selfish nodes. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness. Since traditional replica allocation techniques failed to consider selfish nodes, we also proposed novel replica allocation techniques. Extensive simulation shows that the proposed strategies outperform existing representative cooperative replica allocation techniques in terms of data accessibility, communication cost, and query delay. We are currently working on the impact of different mobility patterns. To identify and handle false alarms in selfish replica allocation.

REFERENCES

[1] E. Adar and B.A. Huberman, "Free Riding on Gnutella," First Monday, vol. 5, no. 10, pp. 1-22, 2000.
[2] L. Anderegg and S. Eidenbenz, "Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents," Proc. ACM MobiCom, pp. 245-259, 2003.
[3] D.J. Baker, J. Wieselthier, and A. Ephremides, "A distributed algorithm for scheduling the activation of links in a self-organizing, mobile, radio network," Proc. IEEE ICC'82, pp.2F6.1-2F6.5, 1982.
[4] K. Balakrishnan, J. Deng, and P.K. Varshney, "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks," Proc. IEEE Wireless Comm. and Networking, pp. 2137-2142, 2005.
[5] R.F. Baumeister and M.R. Leary, "The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation," Psychological Bull., vol. 117, no. 3, pp. 497-529, 1995.
[6] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A performance comparison of multi-Hop wireless ad hoc network routing protocols," Proc. Mobicom'98, pp.159-164, 1992.
[7] D. Hales, "From Selfish Nodes to Cooperative Networks - Emergent Link-Based Incentives in Peer-to-Peer Networks," Proc.IEEE Int'l Conf. Peer-to-Peer Computing, pp. 151-158, 2004.
[8] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," Proc. IEEE INFOCOM, pp. 1568-1576, 2001.
[9] Y. Liu and Y. Yang, "Reputation Propagation and Agreement in Mobile Ad-Hoc Networks," Proc. IEEE Wireless Comm. And Networking Conf., pp. 1510-1515, 2003.
[10] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," Proc. ACM MobiCom, pp. 255-265, 2000.
[11] L.J. Mester, "What's the Point of Credit Scoring?" Business Rev.,pp. 3-16, Sept. 1997.
[12] P. Michiardi and R. Molva, "Simulation-Based Analysis of Security Exposures in Mobile Ad Hoc ," Proc. European Wireless Conf., pp. 1-6, 2002.
[13] M.J. Osborne, An Introduction to Game Theory. Oxford Univ., 2003.
[14] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," Proc. IEEE INFOCOM, pp. 808-817, 2003.
[15] W. Wang, X.-Y. Li, and Y. Wang, "Truthful Multicast Routing in Selfish Wireless Networks," Proc. ACM MobiCom, pp. 245-259, 2004.
[16] Y. Yoo and D.P. Agrawal, "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Comm., vol. 13, no. 6, pp. 87-97, Dec. 2006.